PENERBIT ANDI®

Belajar Sendiri
Pasti Bisa

Javascript

Abdul Kadir

Belajar Sendiri Pasti Bisa

JavaScript

Abdul Kadir

Penerbit ANDI Yogyakarta

Belajar Sendiri Pasti Bisa JavaScript
Oleh: Abdul Kadir

Editor : Th. Arie Prabawati

Setting : Aditya

Desain Cover : Bowo

Korektor : Ratih

Penerbit CV. ANDI OFFSET (Penerbit ANDI)
Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282 Yogyakarta 55281

Percetakan: ANDI OFFSET
Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282 Yogyakarta 55281

Perpustakaan Nasional: Katalog dalam Terbitan (KDT)

Kadir, Abdul

Belajar Sendiri Pasti Bisa JavaScript

Abdul Kadir; – **Ed. I .** – Yogyakarta: ANDI,

24 23 22 21 20 19 18 17 16 15

vi + 338 hlm. ; 20 x 28 Cm.

10 9 8 7 6 5 4 3 2 1

ISBN: 978 – 979 – 29 – 5103 – 5

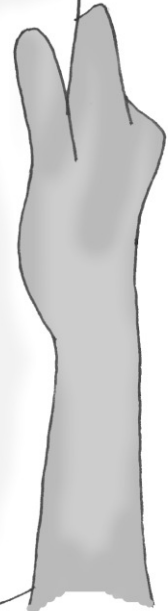
E-ISBN: 978-979-29-8869-7

I. Judul

1. Web Programming

DDC'23 : 006.76

SEKAPUR SIRIH...

Tak lupa, saya ingin menyampaikan pesan dari penyusun modul ini.

Mau tahu isi buku ini?

Berikut adalah 19 modul yang perlu untuk didalami. Isinya pasti akan memperkaya wawasan Anda!

DAFTAR ISI

Sebaiknya ikuti modul secara berurutan.

Selamat berjuang! Mulailah dengan selalu menggaungkan semangat, "Belajar sendiri pasti

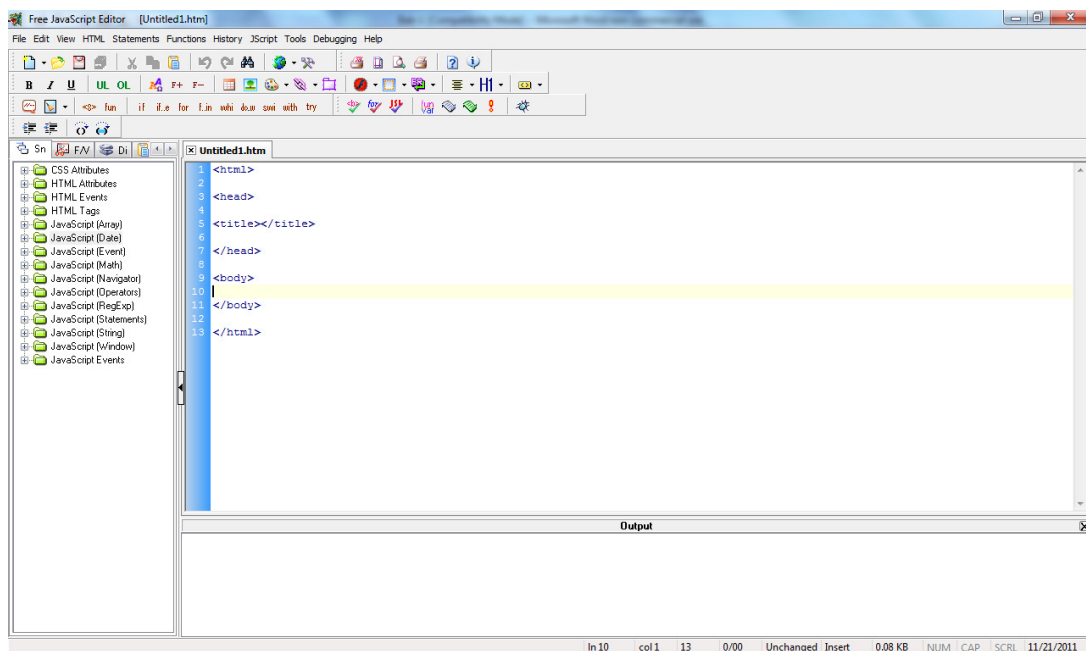
Yuk, Memulai dari Dasar!

Di modul ini, Anda akan belajar mengenal JavaScript. Anda mulai menyiapkan folder-folder untuk latihan dan berlatih hal-hal dasar, termasuk cara penulisan kode dan pencarian kesalahan.

1 Apa yang Perlu Disiapkan Kalau Mau Menggunakan JavaScript?

Anda cukup menggunakan browser seperti Mozilla Firefox atau Internet Explorer. Anda tidak perlu menyiapkan *web server* seperti IIS ataupun Apache. Secara prinsip, tanpa *web server* pun Anda sudah bisa menggunakan JavaScript.

Perlu diketahui, contoh-contoh dalam buku ini menggunakan HTML5. Itulah sebabnya, siapkan browser yang mendukung HTML5. Sebagai contoh, gunakan Firefox 16.x atau Google Chrome.

Untuk mempermudah penyuntingan kode, gunakan editor teks yang mendukung nomor baris. Sebagai contoh, Anda bisa mengunduh Free JavaScript Editor di situs http://www.yaldex.com/Free_JavaScript_Editor.htm. Contoh tampilan editor tersebut seperti berikut.

Gambar 1.1 *Free JavaScript Editor. Versi 4.7 tahun 2009*

Free Java Script Editor ini selain berfungsi sebagai editor juga dapat dimanfaatkan untuk memeriksa kesalahan sintaks kode JavaSript.

2 Dasar apa yang perlu saya miliki kalau mau belajar JavaScript?

Minimal Anda tahu dasar HTML. Materi pada buku ini mengasumsikan bahwa Anda tahu cara menuliskan HTML dan tahu perintah-perintah dasar yang digunakan. Namun, penjelasan singkat tentang tag-tag HTML terkadang dibahas dalam ulasan.

Secara prinsip, struktur dasar kode HTML akan berupa:

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>...</title>
   </head>
   <body>
      ...
   </body>
</html>
```

- Sebuah dokumen HTML5 diawali dengan <!DOCTYPE HTML>.
- Tanda seperti <html> disebut tag. Sebuah tag seperti itu menyatakan sebuah elemen dalam dokumen html.
- Beberapa tag berpasangan. Sebagai contoh, <head> berpasangan dengan </head>.
- Namun, tidak semua tag berpasangan. Sebagai contoh, <br> tidak punya pasangan.
- Pasangan <html>..</html> menyatakan awal dokumen HTML.
- Di dalam <html>..</html> terdapat pasangan <head>..</head> dan <body>..</body>.
- Pasangan <head>..</head> menyatakan bagian judul dokumen HTML. Isinya paling tidak berupa pasangan <title>..</title>.
- Isian yang berada pada <title>..</title> menentukan judul dalam browser.
- Pasangan <body>..</body> menyatakan bagian tubuh dokumen. Bagian ini bisa berisi berbagai tag, misalnya <div>..</div>.

3 Apakah kode JavaScript bisa disembunyikan supaya tidak bisa dibaca orang lain?

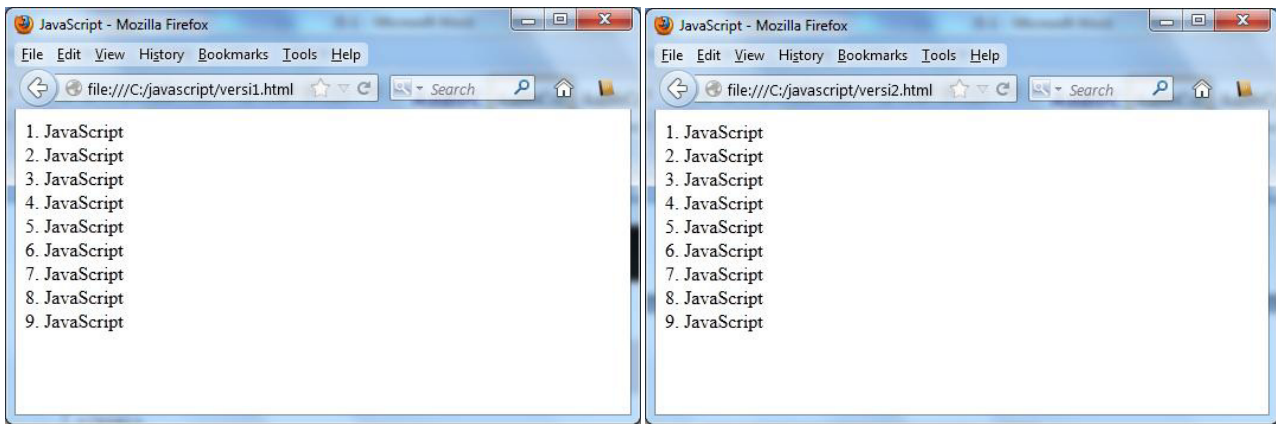
Kode JavaScript bersifat “terbuka”. Artinya, seluruh kode bisa dibaca oleh orang yang membuka situs web. Jadi, tidak ada rahasia yang disembunyikan. Namun, kalau Anda mau, Anda bisa membuat kode Anda menjadi susah dibaca oleh orang. Caranya, buang semua karakter pindah baris dan spasi yang tidak diperlukan. Agar tidak susah membayangkannya, lihat dua contoh berikut.

Versi 1

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>JavaScript</title>
   </head>
   <body>
      <script type="text/javascript">
         var pencacah = 1;

         while (pencacah < 10) {
            document.write(pencacah + ". JavaScript" + "<br>\n");
            pencacah = pencacah + 1;
         }
      </script>
   </body>
</html>
```

Versi 2

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>JavaScript</title>
   </head>
   <body>
      <script type="text/javascript">var
pencacah=1;while(pencacah<10){document.write(pencacah+".
JavaScript"+"<br>\n");pencacah=pencacah+1;}</script>
   </body>
</html>
```

Bagian yang diarsir abu-abu adalah kode JavaScript. Kedua versi penulisan kode JavaScript tersebut memberikan hasil yang sama bagi pemakai. Contoh ditunjukkan di gambar berikut.

Perhatikan bahwa kode JavaScript pada versi pertama lebih mudah kita baca daripada versi kedua. Tentu saja, kode pada versi kedua akan menjadi semakin sulit dibaca kalau kode JavaScript terdiri atas puluhan baris. Walaupun begitu, disarankan untuk menulis kode yang mudah dibaca oleh orang agar mudah dimengerti dan mudah dikembangkan.

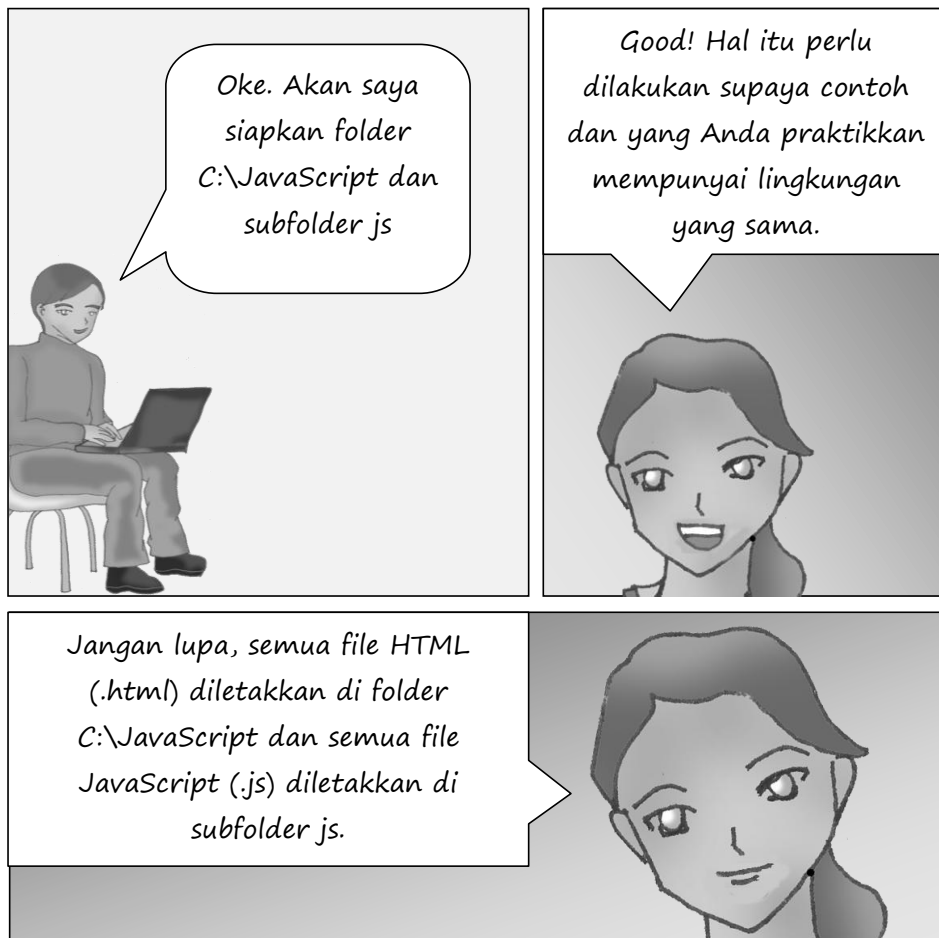
Bagaimana cara untuk mendapatkan hasil seperti di atas?

Anda cukup memanggil nama dokumen HTML di baris alamat (URL) di browser. Sebagai contoh, Anda bisa memanggil `versi1.html` dengan cara menuliskan berikut di baris alamat:
C:\JavaScript\versi1.html

Hal di atas mengasumsikan bahwa file `versi1.html` berada di folder C:\JavaScript. Browser dengan sendirinya akan mengubah ke bentuk semacam:

file:///C:/JavaScript/versi2.html

Catatan

- ✓ **Semua file HTML yang digunakan untuk latihan di buku ini diasumsikan berada di folder `C:\JavaScript`. Oleh karena itu, siapkan folder tersebut terlebih dulu.**
- ✓ **Untuk file-file JavaScript (berekstensi .js), semuanya akan diletakkan di subfolder `js` (di bawah `C:\JavaScript`).**

5 Sebenarnya, di bagian manakah kode JavaScript diletakkan di dokumen HTML?

Umumnya, kode JavaScript diletakkan di dalam pasangan tag <head>..</head> setelah <title>..</title>. Pada contoh di depan, kode JavaScript diletakkan di bagian <body>..<body>. Contoh yang memberikan hasil seperti pada contoh sebelumnya dapat dilihat di skrip berikut.

***File* : versi3.html**

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>JavaScript</title>
      <script type="text/javascript">
         window.onload = init;

         function init() {
            var teks = "";
            var pencacah = 1;
```

```
            while (pencacah < 10) {
               teks = teks + pencacah + ". JavaScript" + "<br>\n";
               pencacah = pencacah + 1;
            }

            var elemenInfo = document.getElementById("info");
            elemenInfo.innerHTML = teks;
         }
      </script>
   </head>
   <body>
      <div id="info"></div>
   </body>
</html>
```

Akhir file

Pada contoh di atas, kode JavaScript terletak di <head>..</head>. Namun, kode JavaScript yang digunakan agak berbeda dengan contoh sebelumnya. Di kode JavaScript terdapat perintah:

```
window.onload = init;
```

Kode tersebut menyatakan bahwa kalau seluruh elemen di HTML sudah termuat di halaman web (dinyatakan dengan atribut `onload` pada `window`), fungsi bernama `init` akan dijalankan. Definisi fungsi `init` adalah seperti berikut:

```
function init() {
   ...
}
```

Di dalam fungsi tersebut, perintah

```
var teks = "";
```

digunakan untuk mendeklarasikan variabel bernama `teks` dan diberi nilai “” (string kosong). String berarti deretan karakter. String kosong berarti string yang tidak mengandung satu karakter pun. Variabel ini akan dipakai untuk menampung tulisan yang dibangkitkan melalui `while`.

Perintah

```
var pencacah = 1;
```

digunakan untuk mendeklarasikan variabel `pencacah` yang mula-mula diisi dengan angka 1. Variabel ini dipakai untuk menghitung dari 1 hingga 9, yang ditangani oleh `for`.

Perintah

```
while (pencacah < 10) {
   teks = teks + pencacah + ". JavaScript" + "<br>\n";
   pencacah = pencacah + 1;
}
```

digunakan untuk membentuk tulisan:

1. JavaScript

hingga

9. JavaScript

dan hasilnya ditampung di variabel `teks`. Tanda + pada pernyataan berikutlah yang berfungsi untuk melakukan penggabungan string:

```
teks = teks + pencacah + ". JavaScript" + "<br>\n";
```

- ✓ **Tag <br> berguna untuk membuat tulisan berikutnya muncul di baris yang berbeda. Efeknya terjadi perpindahan baris di browser.**
- ✓ **Adapun \n menyatakan karakter pindah baris di string dan tidak berefek pada browser. Namun, efek pindah baris akan terlihat kalau perintah `alert()` digunakan.**

Perintah

```
var elemenInfo = document.getElementById("info");
```

digunakan untuk memperoleh elemen HTML yang nilai atribut `id`-nya berupa “info” dan dicatat di `elemenInfo`. Selanjutnya, perintah

```
elemenInfo.innerHTML = teks;
```

membuat elemen yang dirujuk `elemenInfo` diisi dengan isi `teks`. Dengan cara begitu, diperoleh hasil yang sama dengan hasil `versi1.html` maupun `versi2.html`. Perlu diketahui, `innerHTML` menyatakan atribut yang mencatat isi elemen HTML seperti `div`.

6 Bagaimana kalau saya menghendaki kode JavaScript diletakkan terpisah terhadap dokumen html?

Kode JavaScript bisa ditulis pada file berekstensi .js. Sebagai contoh, kode JavaScript berdasarkan `versi3.html` bisa Anda simpan pada file tersendiri dengan nama `versi4.js`. File ini perlu diletakkan di subfolder `js` yang berada di bawah `C:\JavaScript`. Isi file ini seperti berikut:

***File* : versi4.js**

```
window.onload = init;

function init() {
   var teks = "";
   var pencacah = 1;

   while (pencacah < 10) {
      teks = teks + pencacah + ". JavaScript" + "<br>\n";
      pencacah = pencacah + 1;
   }

   var elemenInfo = document.getElementById("info");
   elemenInfo.innerHTML = teks;
}
```

Akhir file

Kode yang perlu diletakkan di file di atas adalah kode yang di `versi3.html` terletak di antara

```
<script type="text/javascript">
```

dan

```
</script>
```

Selanjutnya, lakukan penulisan di file HTML seperti berikut:

***File* : versi4.html**

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>JavaScript</title>
      <script type="text/javascript"
              src="js/versi4.js">
      </script>
   </head>
   <body>
      <div id="info"></div>
   </body>
</html>
```

Akhir file

Perhatikan, tag <script> berisi seperti berikut:

```
<script type="text/javascript"
              src="js/versi4.js">
```

Atribut `src` di tag tersebut menyatakan file JavaScript yang digunakan. Dalam hal ini, file yang dilibatkan adalah file `versi4.js` yang berada di subfolder `js`.

Jika file `versi4.html` ini dipanggil melalui browser, hasilnya akan sama dengan hasil contoh-contoh sebelumnya.

7 Saya mencoba menguji Javascript. Hasilnya tidak seperti yang saya harapkan. Bagaimana saya melacak kesalahan yang terjadi?

Sering kali, hasil yang diharapkan tidak sesuai dengan kenyataan. Sebagai contoh, ketika Anda menguji `versi4.html`, ternyata hasilnya kosong. Tentu, Anda bertanya-tanya, “Kode manakah yang menyebabkan kesalahan?” Salah satu cara yang bisa Anda gunakan adalah memeriksa sintaks kode yang sudah Anda tulis. Anda bisa menggunakan Free JavaScript untuk melakukan hal itu walau tidak sempurna.

Untuk mempraktikkannya, cobalah untuk menyalin isi `versi4.js` dan kemudian menyimpannya ke `versi5.js`. Lalu, ubahlah sehingga menjadi seperti berikut.

File **: versi5.js**

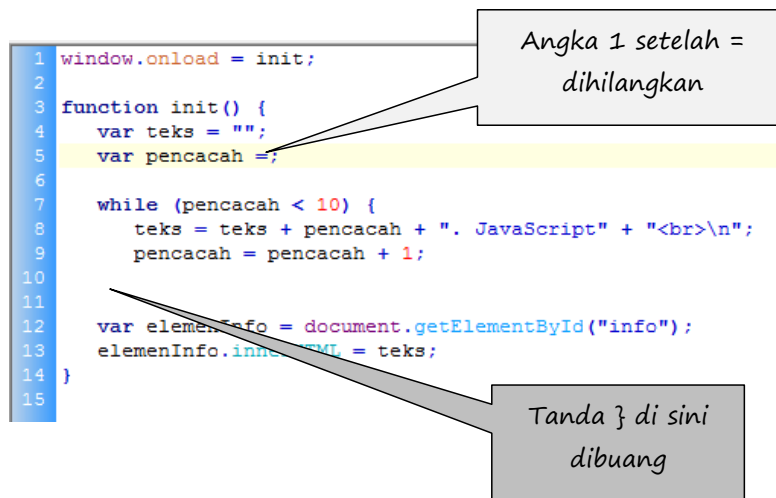
```
window.onload = init;

function init() {
   var teks = "";
   var pencacah  1;

   while (pencacah < 10) {
      teks = teks + pencacah + ". JavaScript" + "<br>\n";
      pencacah = pencacah + 1;


   var elemenInfo = document.getElementById("info");
   elemenInfo.innerHTML = teks;
}
```

Akhir file

Gambar berikut menunjukkan ada dua karakter yang dihapus.

```
1  window.onload = init;
2
3  function init() {
4     var teks = "";
5     var pencacah =;
6
7     while (pencacah < 10) {
8        teks = teks + pencacah + ". JavaScript" + "<br>\n";
9        pencacah = pencacah + 1;
10
11
12    var elemenInfo = document.getElementById("info");
13    elemenInfo.innerHTML = teks;
14 }
15
```

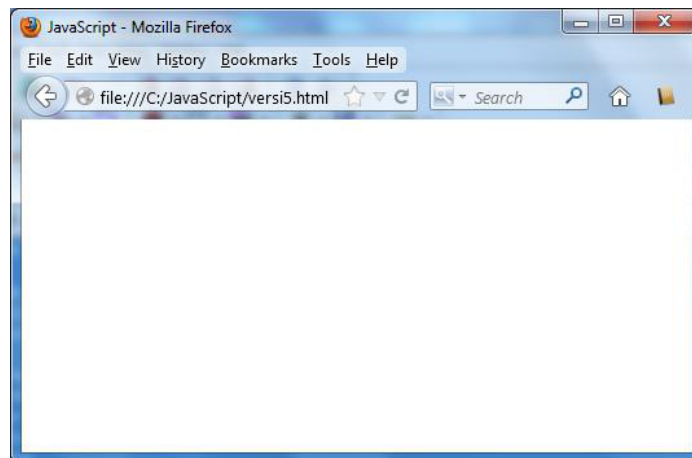
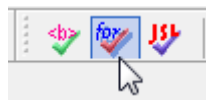
Angka 1 setelah = dihilangkan

Tanda } di sini dibuang

Selanjutnya, salinlah file `versi4.html` menjadi file `versi5.html`. Lalu, gantilah `js/versi4.js` menjadi `js/versi5.js`. Jadi, isinya seperti berikut.

***File* : versi5.html**

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>JavaScript</title>
      <script type="text/javascript"
              src="js/versi5.js">
      </script>
   </head>
   <body>
      <div id="info"></div>
   </body>
</html>
```

Akhir file

Hasilnya adalah seperti berikut:

Kesalahan seperti itu jelas terletak di kode JavaScript. Namun, kode manakah yang menjadi penyebabnya? Untuk mengetahuinya, silakan menuju ke Free JavaScript Editor yang memuat kode `versi5.js`. Lalu, kliklah pada posisi berikut: . Gambar berikut menunjukkan pesan kesalahan di jendela Output.

```
1   window.onload = init;
2
3   function init() {
4      var teks = "";
5      var pencacah =;
6
7      while (pencacah < 10) {
8         teks = teks + pencacah + ". JavaScript" + "<br>\n";
9         pencacah = pencacah + 1;
10
11
12     var elemenInfo = document.getElementById("info");
13     elemenInfo.innerHTML = teks;
14  }
15
```

Output

Line 5 Column 18 Error: Identifier expected

Pesan di atas menyatakan bahwa baris 5 kolom 18 ada yang salah. Kesalahannya adalah karena ada nilai yang perlu diberikan sesudah tanda sama dengan. Sekarang, berdasarkan keterangan tersebut, Anda bisa menambahkan angka 1 sesudah tanda =.

Setelah penambahan angka 1 tersebut, klik kembali pada posisi .

Pesan kesalahan sekarang berupa:

```
1  window.onload = init;
2
3  function init() {
4     var teks = ""
5     var pencacah = 1;
6
7     while (pencacah < 10) {
8        teks = teks + pencacah + ". JavaScript" + "<br>\n";
9        pencacah = pencacah + 1;
10
11
12    var elemenInfo = document.getElementById("info");
13    elemenInfo.innerHTML = teks;
14 }
15
```

Output

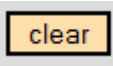
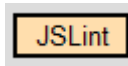
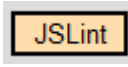
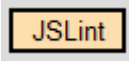
Line 15 Column 1 Error: } expected

Pesan kesalahan di atas menyatakan bahwa di baris 15 terdapat kekurangan }. Namun, tentu saja, belum tentu kekurangan } tersebut terletak di baris 15. Sebagai contoh, berdasarkan perintah itu, Anda mengetahui posisi yang membutuhkan } adalah di bawah

pencacah = {pencacah + 1Tambahkan } pada posisi tersebut.

Setelah penambahan tanda } tersebut, klik kembali pada posisi . Pesan yang terjadi berupa: There is no JavaScript error . Hal itu menyatakan bahwa sudah tidak ada kesalahan lagi.

Catatan

Walaupun *Free JavaScript Editor* mengeluarkan There is no JavaScript error **pesan, belum tentu tidak ada kesalahan secara sintaks di kode JavaScript.**

Fasilitas lain untuk memverifikasi kode JavaScript, yaitu JSLint. Pemverifasi JavaScript buatan Douglas Crockford ini tersedia di Free JavaScript Editor. Supaya Anda bisa memanfaatkannya, perlu ada sedikit perubahan yang perlu dilakukan karena tampaknya ada *bug*. Hal yang perlu Anda lakukan sekali adalah seperti berikut.

1. Dengan menggunakan Windows Explorer, pilihlah folder `C:\Program Files\Yaldex Software\Free JavaScript Editor\JSInt`. Di folder tersebut terdapat file bernama `JSLints`.
2. Gantilah namanya menjadi `JSLint`.

Untuk melakukan pengujian dengan JSLint, lakukan langkah seperti berikut.

1. Klik pada posisi berikut: . Langkah ini akan menampilkan halaman web seperti berikut:

2. Seleksilah kode HTML yang mengandung JavaScript atau kode JavaScript saja dan lakukan penyalinan (dengan Ctrl+C).
3. Klik di tombol clear untuk mengosongkan isi kotak yang berada di atas tombol JSLint.
4. Lakukan Ctrl+V di kotak yang berada di atas tombol JSLint.
5. Klik di tombol JSLint untuk melakukan verifikasi.

Catatan

✓ **JSLint terkadang menyatakan kode yang benar dianggap salah karena suatu aturan ketat yang diterapkan di JSLint ketika verifikasi dilakukan. Oleh karena itu, kadang-kadang Anda perlu mengabaikan pesan kesalahan tersebut dan langsung mengujinya di browser saja.**

Untuk mencoba JSLint, modifkasilah `versi5.js` menjadi seperti berikut:

```
1  window.onload = init;
2
3  function init() {
4      var teks = "";
5      var pencacah  1;
6
7      while (pencacah < 10) {
8          teks = teks + pencacah + ". JavaScript" + "<br>\n";
9          pencacah = pencacah + 1;
10     }
11
12     var elemenInfo = document.getElementById("info");
13     elemenInfo.innerHTML = teks;
14 }
```

Tanda = dihilangkan

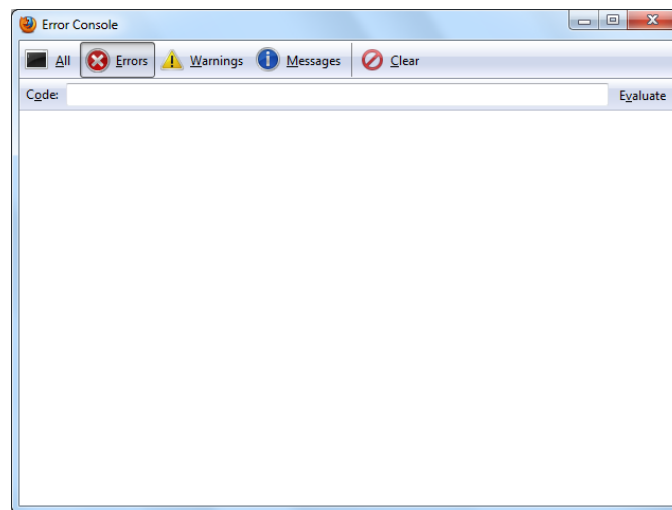
Jika hanya menggunakan JavaScript Editor, kesalahan seperti di atas tidak terdeteksi. Namun, kalau menggunakan JSLint, kesalahannya seperti berikut:

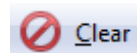
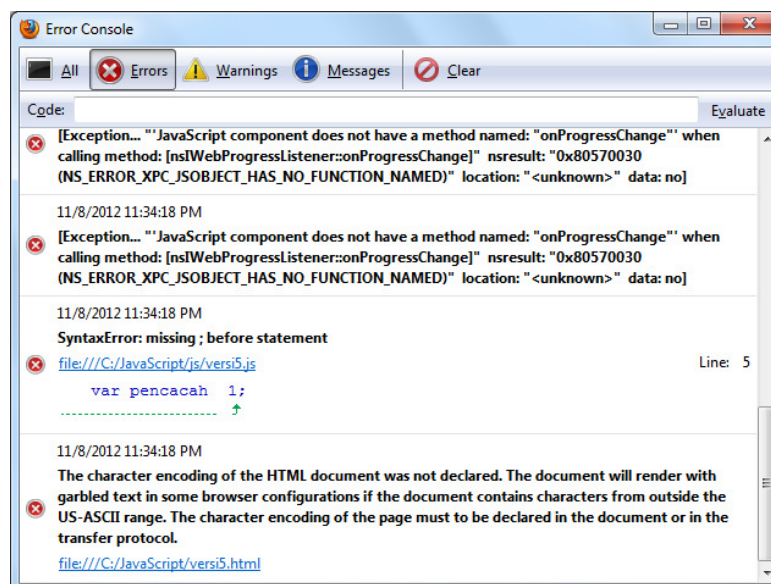
```
Error:
    Problem at line 5 character 16: Missing semicolon.
    var pencacah 1;
    Problem at line 5 character 18: Expected an assignment or function call and instead saw an expression.
    var pencacah 1;
```

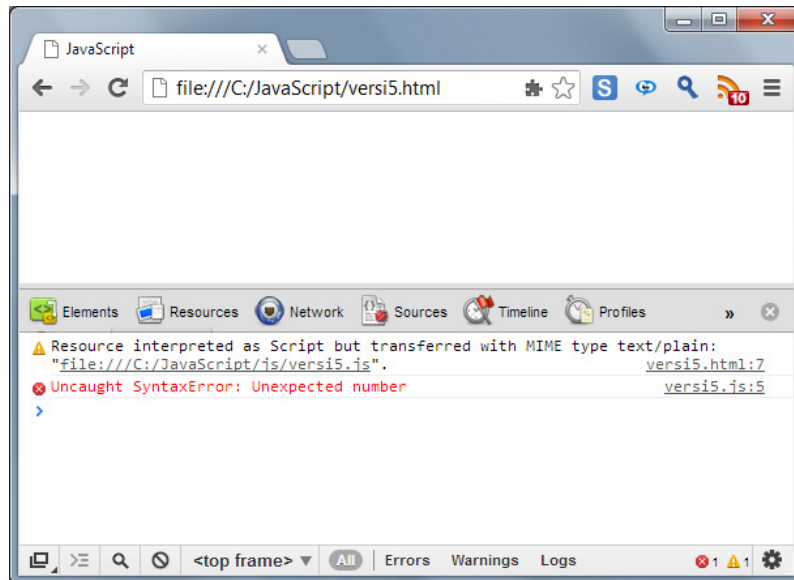
Kesalahannya terletak di baris 5 karakter ke-18. Solusi yang diberikan berupa tanda sama dengan atau berupa pemanggilan fungsi. Petunjuk tersebut tentu saja berguna di dalam melakukan koreksi.

Jika Anda menggunakan browser Firefox, kesalahan pada kode JavaScript sebenarnya bisa dilihat di *Error Console*. Untuk mengaktifkan *Error Console*, lakukan langkah-langkah berikut.

1. Klik pada menu **Tools** di browser Firefox.
2. Letakkan penunjuk mouse di Web Developer.
3. Klik pada Error Console.

Pada keadaan awal, tampilan *Error Console* adalah seperti berikut:

Tampilan seperti itu bisa juga diperoleh setelah dilakukan pengklikan pada

.

Sekarang, Anda bisa mencoba memanggil `versi5.html` di browser. Hasil di *Error Console* menjadi seperti berikut:

Informasi terpenting yang perlu kita perhatikan adalah pada *SyntaxError*. Saran yang diberikan adalah memberikan tanda ; sebelum angka 1. Walaupun solusinya tidak seperti itu, pesan tersebut paling tidak memberikan arahan lokasi yang salah.

Jika Anda menggunakan Google Chrome, Anda bisa memantau kesalahan dengan menekan Ctrl+Shift+J terlebih dulu. Kemudian, Anda bisa mencoba dokumen yang akan diuji. Berikut adalah contoh hasil pemanggilan `versi5.html`:

Pada contoh di atas, kesalahan yang terjadi berupa: *Uncaught SyntaxError: unexpected number*. Pesan ini menyatakan ada angka yang tidak diharapkan. Kesalahan dinyatakan dalam `versi5.html:5`. Artinya, kesalahan terdapat di file `versi5.html` baris 5. Seperti halnya di Firefox, pesan kesalahan dapat menjadi pedoman untuk membetulkan kode di lokasi yang ditunjukkan.

8 *Secara sintaks, kode yang saya tulis sudah benar. Namun, hasilnya tidak sesuai dengan yang saya harapkan. Apa yang harus dilakukan?*

Kesalahan seperti itu dinamakan sebagai kesalahan logika. Tentu saja, kesalahan logika tidak terdeteksi oleh browser ataupun tool apa pun. Untuk memahami kesalahan logika, lihatlah contoh berikut:

***File* : keliling.html**

```
<!DOCTYPE HTML>
<html>
   <head>
      <title>JavaScript</title>
      <script type="text/javascript"
```